CMPSCI 240: "Reasoning Under Uncertainty"

Lecture 11

Prof. Hanna Wallach
wallach@cs.umass.edu

February 28, 2012

Reminders

- Check the course website: `http://www.cs.umass.edu/~wallach/courses/s12/cmpsci240/`
- Fourth homework is due on Friday

Recap

Last Time: Representing Events

- Fixed length codes: use same number of bits to encode each event, e.g., $A_1 = 11$, $A_2 = 10$, $A_3 = 01$, $A_4 = 00$

Last Time: Representing Events

- Fixed length codes: use same number of bits to encode each event, e.g., $A_1 = 11$, $A_2 = 10$, $A_3 = 01$, $A_4 = 00$
- Optimal for events with equal probabilities

Last Time: Representing Events

- Fixed length codes: use same number of bits to encode each event, e.g., $A_1 = 11$, $A_2 = 10$, $A_3 = 01$, $A_4 = 00$
- Optimal for events with equal probabilities
- Variable length codes: use different number of bits to encode each event, e.g., $A_1 = 1$, $A_2 = 01$, $A_3 = 001$, $A_4 = 000$

Last Time: Representing Events

- Fixed length codes: use same number of bits to encode each event, e.g., $A_1 = 11$, $A_2 = 10$, $A_3 = 01$, $A_4 = 00$
- Optimal for events with equal probabilities
- Variable length codes: use different number of bits to encode each event, e.g., $A_1 = 1$, $A_2 = 01$, $A_3 = 001$, $A_4 = 000$
- Optimal for events with unequal probabilities

Last Time: Prefix Codes

- Prefix code: no "code word" is a prefix of any other

Last Time: Prefix Codes

- Prefix code: no “code word” is a prefix of any other
- Messages (sequences of events) can be uniquely decoded

Last Time: Prefix Codes

- Prefix code: no “code word” is a prefix of any other
- Messages (sequences of events) can be uniquely decoded
- Messages are easy to decode in a left-to-right fashion

Last Time: Prefix Codes

- Prefix code: no "code word" is a prefix of any other
- Messages (sequences of events) can be uniquely decoded
- Messages are easy to decode in a left-to-right fashion
- Any binary tree with events as leaves defines a prefix code

Last Time: Prefix Codes

- Prefix code: no "code word" is a prefix of any other
- Messages (sequences of events) can be uniquely decoded
- Messages are easy to decode in a left-to-right fashion
- Any binary tree with events as leaves defines a prefix code
- Not necessarily optimal (information rate $\geq$ entropy)

Last Time: Huffman Coding

- Bottom-up construction: build the tree from the leaves up

Last Time: Huffman Coding

- Bottom-up construction: build the tree from the leaves up
- Merge two least probable events (leaves or subtrees)

Last Time: Huffman Coding

- Bottom-up construction: build the tree from the leaves up
- Merge two least probable events (leaves or subtrees)

- Upper bound on information rate is better:

$$H(A_1, \ldots, A_n) \leq R(A_1, \ldots, A_n) < H(A_1, \ldots, A_n) + 1$$

Last Time: Huffman Coding

- Bottom-up construction: build the tree from the leaves up
- Merge two least probable events (leaves or subtrees)
- Upper bound on information rate is better:

$$H(A_1, \ldots, A_n) \leq R(A_1, \ldots, A_n) < H(A_1, \ldots, A_n) + 1$$

- Can prove this is optimal for a prefix code

Examples of Huffman Coding

	A_1	A_2	A_3	A_4	A_5
$P(A_i)$	0.25	0.25	0.2	0.15	0.15

Examples of Huffman Coding

	A_1	A_2	A_3	A_4	A_5
$P(A_i)$	0.25	0.25	0.2	0.15	0.15

- e.g., what is the information content of each event?

Examples of Huffman Coding

	A_1	A_2	A_3	A_4	A_5
$P(A_i)$	0.25	0.25	0.2	0.15	0.15

- e.g., what is the information content of each event?
- e.g., what is the information rate?

Examples of Huffman Coding

	A_1	A_2	A_3	A_4	A_5
$P(A_i)$	0.25	0.25	0.2	0.15	0.15

- e.g., what is the information content of each event?
- e.g., what is the information rate?
- e.g., what is the entropy?

Communicating Perfectly

Transmitting Information

- Goal: transmit some message, encoded in binary

Transmitting Information

- Goal: transmit some message, encoded in binary
- Want to make sure the correct message is received even if there are transmission errors (e.g., static, disk failure, ...)

Transmitting Information

- Goal: transmit some message, encoded in binary
- Want to make sure the correct message is received even if there are transmission errors (e.g., static, disk failure, ...)
- Probability of a single bit being flipped is p

Transmitting Information

- Goal: transmit some message, encoded in binary
- Want to make sure the correct message is received even if there are transmission errors (e.g., static, disk failure, ...)
- Probability of a single bit being flipped is p
- Error probability: overall probability of there being an undetected error when using some encoding scheme

Examples of Error Probability

- e.g., 8 events represented as 000, 001, 010, ... 111, probability of a single bit flip is $1 / 10$, what is the error probability?

Encoding with Redundancy

- Can use additional bits when encoding events to ensure that they are "protected" against errors in transmission

Encoding with Redundancy

- Can use additional bits when encoding events to ensure that they are “protected” against errors in transmission
- Error detecting codes vs. error correcting codes

Encoding with Redundancy

- Can use additional bits when encoding events to ensure that they are “protected” against errors in transmission
- Error detecting codes vs. error correcting codes
- Fundamental trade-off: want encoding schemes that minimize both the error probability and the information rate

Error-Detecting Codes: Parity Check Codes

- Append a parity bit to each code word such that every code word always contains an even number of ones

Error-Detecting Codes: Parity Check Codes

- Append a parity bit to each code word such that every code word always contains an even number of ones
- e.g., 0000, 0011, 0101, . . . , 1100, 1111

Error-Detecting Codes: Parity Check Codes

- Append a parity bit to each code word such that every code word always contains an even number of ones
- e.g., 0000, 0011, 0101, . . . , 1100, 1111
- Can detect error if an odd number of bits get flipped

Error-Detecting Codes: Parity Check Codes

- Append a parity bit to each code word such that every code word always contains an even number of ones
- e.g., 0000, 0011, 0101, . . . , 1100, 1111
- Can detect error if an odd number of bits get flipped
- Cannot detect error if an even number of bits get flipped

Examples of Parity Check Codes

- e.g., 8 events represented as 0000, 0011, ..., 1111, probability of a single bit flip is $1\,/\,10$, what is the error probability?

Hamming Distance

- Hamming distance: number of positions (bits) in which two binary strings of equal length differ from each other

Hamming Distance

- Hamming distance: number of positions (bits) in which two binary strings of equal length differ from each other
- Adding a parity bit means that any two code words have a Hamming distance of at least 2 from each other

Hamming Distance

- Hamming distance: number of positions (bits) in which two binary strings of equal length differ from each other
- Adding a parity bit means that any two code words have a Hamming distance of at least 2 from each other
- e.g., 000 and 001 vs. 0000 and 0011

Hamming Distance

- Hamming distance: number of positions (bits) in which two binary strings of equal length differ from each other
- Adding a parity bit means that any two code words have a Hamming distance of at least 2 from each other
- e.g., 000 and 001 vs. 0000 and 0011
- Can only detect odd number of bit flips, can't correct errors

Error-Correcting Codes: 7/4 Hamming Codes

- e.g., 16 events represented as 0000, 0001, . . . , 1111

Error-Correcting Codes: 7/4 Hamming Codes

- e.g., 16 events represented as 0000, 0001, . . . , 1111
- Add 3 bits $t_5 t_6 t_7$ to each code word $s_1 s_2 s_3 s_4$ such that

$$t_5 = s_1 + s_2 + s_3 \pmod 2$$
$$t_6 = s_2 + s_3 + s_4 \pmod 2$$
$$t_7 = s_3 + s_4 + s_1 \pmod 2$$

Error-Correcting Codes: 7/4 Hamming Codes

- e.g., 16 events represented as 0000, 0001, . . . , 1111
- Add 3 bits $t_5 t_6 t_7$ to each code word $s_1 s_2 s_3 s_4$ such that

$$\begin{aligned} t_5 &= s_1 + s_2 + s_3 \pmod 2 \\ t_6 &= s_2 + s_3 + s_4 \pmod 2 \\ t_7 &= s_3 + s_4 + s_1 \pmod 2 \end{aligned}$$

- e.g., what do 0000, 0001, . . . , 0101, . . . , 1111 become?

Error-Correcting Codes: 7/4 Hamming Codes

- Any two code words have a Hamming distance of ≥ 3

Error-Correcting Codes: 7/4 Hamming Codes

- Any two code words have a Hamming distance of ≥ 3
- 1 bit flip can be detected and corrected

Error-Correcting Codes: 7/4 Hamming Codes

- Any two code words have a Hamming distance of ≥ 3
- 1 bit flip can be detected and corrected
- ≥ 2 bit flips will be corrected to the wrong code word

Error-Correcting Codes: 7/4 Hamming Codes

- Any two code words have a Hamming distance of ≥ 3

- 1 bit flip can be detected and corrected
- ≥ 2 bit flips will be corrected to the wrong code word

- 2 bit flips can be detected using a global parity bit $\implies 8/4$

Correcting Single-Bit Errors

- Write the received code word in 3 overlapping circles

Correcting Single-Bit Errors

- Write the received code word in 3 overlapping circles
- Goal: every circle should have parity 0 (i.e., even # 1s)

Correcting Single-Bit Errors

- Write the received code word in 3 overlapping circles
- Goal: every circle should have parity 0 (i.e., even # 1s)
- Check each circle to see if its parity is 0 or 1

Correcting Single-Bit Errors

- Write the received code word in 3 overlapping circles
- Goal: every circle should have parity 0 (i.e., even # 1s)
- Check each circle to see if its parity is 0 or 1
- Is there a single unique bit (s or t) that lies inside all the parity 1 circles but outside all the parity 0 circles?

Correcting Single-Bit Errors

- Write the received code word in 3 overlapping circles
- Goal: every circle should have parity 0 (i.e., even # 1s)
- Check each circle to see if its parity is 0 or 1
- Is there a single unique bit (s or t) that lies inside all the parity 1 circles but outside all the parity 0 circles?
- If so, flipping this bit accounts for the parity violation

Examples of Decoding 7/4 Hamming Codes

- e.g., suppose 1000101 was transmitted but a) $1000\underline{0}01$, b) $1\underline{1}00101$, c) $10\underline{1}0101$, d) $10\underline{1}010\underline{0}$ were received?

Examples of Error Probability

- e.g., 16 events represented as 0000, 0001, . . . 1111, probability of a single bit flip is $1 \,/\, 10$, what is the error probability?

Examples of Error Probability

- e.g., 16 events now represented as 0000000, 0001011, . . . , 1111111, now what is the error probability?

For Next Time

- Check the course website: `http://www.cs.umass.edu/~wallach/courses/s12/cmpsci240/`
- Fourth homework is due on Friday